Elasticsearch

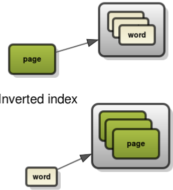
БГУ ФПМИ, 2017

Elasticsearch

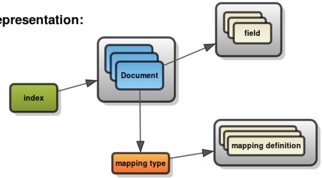
- open-source distributed search server
- implementation is in Java
- powered by **Apache Lucene**
- to achieve fast search responses, instead of searching the text directly, it searches an **index** instead.

Elasticsearch

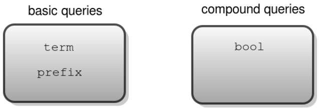
Indexing:



Data representation:



Query DSL:



main query structure

```
curl -X POST "http://localhost:9200/blog/_search?pretty=true" -d '{
                        "from": 0,
                        "size": 10,
                     "query" : QUERY_JSON,
                        FILTER_JSON,
                        FACET_JSON,
                         SORT_JSON
                            }'
```

Structure

```
http://hostname:port/index_name/doc_type/doc_id
```

- An **index** consists of one or more **documents**, and a **document** consists of one or more **fields**.
- A **type** is a logical category/partition of your index whose semantics is completely up to you.

Concepts

Near Realtime

There is a slight latency (normally one second) from the time you index a document until the time it becomes searchable.

Shards & Replicas

Each index can be split into multiple shards (default=5). An index can also be replicated zero or more times (default=1).

Install

```
wget https://artifacts.elastic.co/downloads/elasticsearch/
elasticsearch-5.2.1.tar.gz

tar -zxvf elasticsearch-5.2.1.tar.gz

elasticsearch-5.2.1/bin/elasticsearch

sudo pip3 install elasticsearch
```

Run

```
./elasticsearch -Ecluster.name=c_name -Enode.name=n_name
```

```
http://localhost:9200/
```

```
{
  "name" : "XHJP8FD",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "LbmdPlaRTmKVbULaI6fmUg",
  "version" : {
    "number" : "5.2.1",
    "build_hash" : "db0d481",
    "build_date" : "2017-02-09T22:05:32.386Z",
    "build_snapshot" : false,
    "lucene_version" : "6.4.1"
  },
  "tagline" : "You Know, for Search"
}
```

Search Basics

Create Index

```
curl -XPUT 'localhost:9200/fox_ind' -d '{
"settings": {
  "index.number_of_replicas": 0,
  "index.number_of_shards": 1
}}'
```

Let's Index!

```
curl -XPUT http://localhost:9200/fox_ind/texts/1 -d '{
    "content": "The quick brown fox"
}'
curl -XPUT http://localhost:9200/fox_ind/texts/2 -d '{
    "content": "Jumped over the lazy dog"
}'
curl -XPUT http://localhost:9200/fox_ind/texts/3 -d '{
    "content": "What does the fox say?"
}'
curl -XPUT http://localhost:9200/fox_ind/texts/4 -d '{
    "content": "The quick lazy brown fox did not jump."
}'
```

Index in batch

```
curl -XPOST http://localhost:9200/fox_ind/texts/_bulk?pretty -d
'{"index": {"_id":"1"}}
{"content": "The quick brown fox"}
{"index": {"_id":"2"}}
{"content": "Jumped over the lazy dog"}
{"index": {"_id":"3"}}
{"content": "What does the fox say?"}
{"index": {"_id":"4"}}
{"content": "The quick lazy brown fox did not jump."}'
```

GET /_cat/

```
curl -XGET 'localhost:9200/_cat/health?v'

curl -XGET 'localhost:9200/_cat/nodes?v'

curl -XGET 'localhost:9200/_cat/indices?v'

curl -XGET 'localhost:9200/_cat/shards?v'
```

GET /_stats/

```
curl -XGET 'localhost:9200/fox_ind/_stats/docs?pretty=true'

curl -XGET 'localhost:9200/fox_ind/_stats/store?pretty=true
&human=true'
```

Shrink Index

```
curl -XPUT 'localhost:9200/fox_ind/_settings' -d '{
"settings": {
  "index.blocks.write": true
}}'

curl -XPOST 'localhost:9200/fox_ind/_shrink/fox_ind_shr' -d '{
"settings": {
  "index.number_of_replicas": 0,
  "index.number_of_shards": 1
}}'
```

GET Document

Realtime operation:

```
curl -XGET 'localhost:9200/fox_ind/texts/1?pretty'
```

```
{
  "_index" : "fox_ind",
  "_type" : "texts",
  "_id" : "1",
  "_version" : 1,
  "found" : true,
  "_source" : {
    "content" : "The quick brown fox"
  }
}
```

Simple Search

```
curl -XGET 'localhost:9200/fox_ind/_search?q=fox&pretty=true'
```

```
{
  "took" : 13,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "failed" : 0
  },
  "hits" : {
    "total" : 3,
    "max_score" : 0.561078,
    "hits" : [
      {
        "_index" : "fox_ind",
        "_type" : "texts",
        "_id" : "4",
        "_score" : 0.561078,
        "_source" : {
          "content" : "The quick lazy brown fox did not jump."
        }
      },
     ...
     ]
    }
}
```

Boolean Search

"fox" AND "dog"

```
curl -XGET 'localhost:9200/fox_ind/_search?pretty' -d '{
"query": {
  "bool": {
    "must": [
      {"match": {"content": "dog"} },
      {"match": {"content": "fox"} }
    ]
  }
}
}'
```

"fox" OR "dog"

```
curl -XGET 'localhost:9200/fox_ind/_search?pretty' -d '{
"query": {
  "bool": {
    "should": [
      {"match": {"content": "dog"} },
      {"match": {"content": "fox"} }
    ]
  }
}
}'
```

Fuzzy Match

```
curl -XPOST 'localhost:9200/fox_ind/_search?pretty' -d '{
  "query": {
    "match": {
      "content": {"query": "lary fix", "fuzziness": 1,
                  "operator": "AND"}
    }
  }
}'
```

"fuzziness": "AUTO" is recommended (fuzziness depends on term length).

TF-IDF

Term vectors

Для конкретного документа:

```
curl -XGET 'localhost:9200/fox_ind/texts/1/_termvectors?pretty' -d '{
"fields": ["content"],
"offsets": false,
"payloads": false,
"positions": false,
"term_statistics": true,
"field_statistics": true
}'
```

Статистика для терма по шарду:

```
curl -XGET 'localhost:9200/fox_ind/texts/_termvectors?pretty' -d '{
"doc": {"content": "brown"},
"term_statistics" : true, "positions" : false, "offsets" : false}'
```

Term vectors: top-3 keywords

```
curl -XGET 'localhost:9200/fox_ind/texts/4/_termvectors?pretty'
-d '{
"fields": ["content"],
"term_statistics" : true,
"offsets": false,
"positions": false,
"field_statistics": false,
"filter" : {"max_num_terms" : 3}
}'
```

Understanding _score

Lucene (and thus Elasticsearch) uses the Boolean model to find matching documents, and a formula called the **practical scoring function** to calculate relevance. This formula borrows concepts from tf-idf and the vector space model.

```
curl -XGET 'localhost:9200/fox_ind/texts/_search?q=fox&pretty&explain'
```

В версии 5.2.1 увидите стандартный BM25 (по-умолчанию без boost-добавок).

Changing defaults

```
curl -XPOST 'localhost:9200/fox_ind/_close'

curl -XPUT 'localhost:9200/fox_ind/_settings?pretty' -d '{
"settings": {
  "index": {
    "similarity": { "default": {
      "type": "BM25",
      "b": 0,
      "k1": 2
    }}
  }
}}'

curl -XPOST 'localhost:9200/fox_ind/_open'
```

Другие similarity

- BM25
- Divergence from random
- TF-IDF
- ...

`https://www.elastic.co/guide/en/elasticsearch/reference/master/index-modules-similarity.html#_available_similarities`

Analyzer

```
curl -XGET ’localhost:9200/fox_ind/_analyze?pretty’ -d ’
{"analyzer": "english", "text": "Text was semi-analyzed"}’
```

Analyzer

```
{
  "settings": {
    "analysis": {
      "filter": {
        "russian_stop": {
          "type":       "stop",
          "stopwords":  "_russian_"
        },
        "russian_stemmer": {
          "type":       "stemmer",
          "language":   "russian"
        }
      },
      "analyzer": {
        "russian": {
          "tokenizer":  "standard",
          "filter": [
            "lowercase",
            "russian_stop",
```